

CONTROLADORES DIFUSOS EN MICROCONTROLADORES: SOFTWARE PARA DISEÑO E IMPLEMENTACIÓN

CONTROLLERS FUZZY IN MICROCONTROLLERS: SOFTWARE FOR DESIGN AND IMPLEMENTATION

JOHAN CALDERÓN ACERO¹
IVONNE VANESA PARRA GARZÓN²
HENRY MONTAÑA QUINTERO³

RECIBIDO: MARZO 2010
APROBADO: AGOSTO 2010

RESUMEN

en este artículo se muestra el desarrollo, la implementación y los resultados obtenidos en la construcción de una herramienta para el análisis, diseño e implementación de sistemas de lógica difusa en sistemas microcontrolados, microprocesados, o procesadores con compilador compatible con ANSI C, denominado FUZZY UDTECNO. La herramienta realiza la verificación de los controladores diseñados en *software* de simulación. El programa permite el diseño de controladores o la utilización de sistemas de inferencia difusa en diferentes aplicaciones, lo cual facilita la implementación, puesto que no hay necesidad de tener un profundo conocimiento de programación de dispositivos. Así, se convierte en un instrumento para el aprendizaje y la popularización de la teoría difusa a nivel técnico, tecnológico e ingenieril.

Palabras claves

lógica difusa, microcontroladores, implementación controladores, controlador difuso

Abstract

This article shows the work and results obtained in the development of a tool for the analysis, design and implementation fuzzy logic systems in microcontroller systems,

microprocessor or other processor compatible with ANSI C compiler, FUZZY UDTECNO called, in addition to the systems courses fuzzy concepts regarding verification and as tool control strategies in simulation software. Software allows the design of controllers or use of systems fuzzy inference beyond the control applications, facilitating the implementation without need for pro-

1. Tecnólogo en Electrónica. ODMEL Ltda. Correo electrónico: nahhoj@hotmail.com

2. Tecnóloga en Electrónica. Universidad Distrital Francisco José de Caldas. Correo electrónico: vanjoh_18@hotmail.com

3. Ingeniero electrónico. Docente, Universidad Distrital Francisco José de Caldas. Correo electrónico: hmontana@gmail.com / hmontanaq@udistrital.edu.co

gramming knowledge devices, so becoming a tool for learning and the widespread popularization of the theory at the technical, technological and engineering.

Key words

fuzzy logic, microcontrollers, implementation, controller, inference

INTRODUCCIÓN

En la década de los noventa del siglo xx, el control difuso ganó gran aceptación en la industria como un maduro desarrollo académico. El éxito mundial de innumerables productos comerciales y aplicaciones, ha probado que la tecnología difusa no sólo es práctica y poderosa, sino también altamente beneficiosa y efectiva respecto al costo de su aplicación. Esto se debe, entre otras causas, a que los sistemas del mundo real son no lineales y su modelación exacta es difícil, costosa, y en muchos casos imposible. Entre tanto, el control difuso tiene la potencialidad única para aplicarse a la realización de tareas, sin conocer el modelo del sistema, aun si dichas tareas obedecen a patrones no lineales y complejos [1].

En el marco académico, la verificación de los controladores difusos se realiza por medio de *software* de simulación. Estos controladores muestran resultados ideales y no permiten, además, ver de manera real el comportamiento de los diseños. Para el estudio del control difuso no hay versatilidad para emular diferentes modelos de sistemas ni tampoco para probar los controladores.

Considerando lo anterior, se tomó la decisión de desarrollar un *software* para la implementación de controladores difusos, con el fin de facilitar al educando los procesos

de ejemplificación en la práctica de los conceptos teóricos, de tal manera que agilice su adquisición de conocimientos y permita la apropiación y la transferencia de esta tecnología sin necesidad de tener conocimientos muy avanzados de ella.

1. DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

La finalidad del *software* es el desarrollo de sistemas y controladores difusos y la generación del código de estos mismos para microcontroladores o sistemas microprocesados. Para la implementación de estos sistemas se requiere desarrollar interfaces mínimas como: configuración de la resolución a trabajar, manejo de conjuntos para cada universo –tanto de entradas como de salidas–, interfaz de base de reglas, interfaz de inferencia y generación del código para microcontroladores, entre otras.

El sistema permite a los usuarios diseñar cada uno de los conjuntos difusos, de acuerdo con la configuración de los universos de entrada, y puede operar con cada uno de ellos definiendo una base de reglas fundamentadas en el juicio del experto. También se muestra el proceso de inferencia difusa, y por último, su valor de defusificación, todo esto en relación con la configuración de características del controlador que debe realizarse previamente. El sistema es abierto y escalable, por lo que permite la incorporación de nuevas entradas, conjuntos y reglas, conforme a las limitaciones contempladas en el diseño.

Se trabajó con un lenguaje de programación orientado a objetos, VB.NET®, debido a que el entorno es gráfico. Además, las herramientas de desarrollo incluidas por Microsoft en su Visual Studio.Net® son mucho

más simples, intuitivas y sencillas de manejar que las herramientas de desarrollo equivalentes en JAVA y en otros lenguajes de programación.

- Permite el diseño de bases de reglas completas o incompletas.
- Provee herramientas de diseño rápido para los universos de entrada y salida,

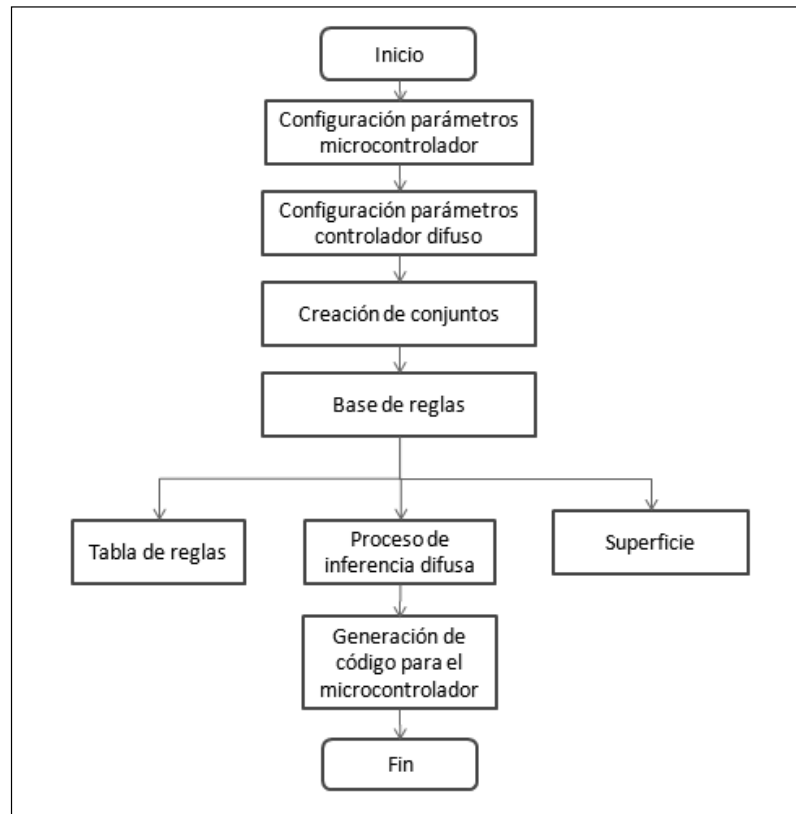


Figura 1. Diagrama general del programa Fuzzy Udtectno

En el diagrama de flujo de la figura 1 se presenta el esquema general de los componentes del programa:

Las principales características del software Fuzzy Udtectno son las siguientes:

- Permite configurar los parámetros del controlador difuso.
- Permite el diseño gráfico de los universos de entrada y salida.

las variables lingüísticas y la base de reglas.

- Permite analizar el comportamiento global del sistema, mediante tablas entrada-salida, y la visualización de la superficie difusa.
- Permite analizar el comportamiento del sistema a entradas particulares, a través de la ventana de simulación.
- Provee el código fuente ANSI C, C18, C30 y .fis del sistema diseñado.

La configuración de parámetros del microcontrolador permite seleccionar el número de entradas y la resolución deseada, mientras que la configuración del controlador difuso permite seleccionar el tipo de implicación (mínimo o producto), agregación (máximo o suma) y defusificación (centro de gravedad, promedio de máximos, primer o último máximo) para el controlador tipo Mamdani.

Para la creación de conjuntos se definieron cuatro funciones de pertenencia: triangular, trapezoidal, sigmoidea y gaussiana. Insertando y seleccionando cada una de ellas se pueden modificar sus parámetros. Entre mayor sea la cantidad de conjuntos que se elija, mayores serán el poder descriptivo y

la flexibilidad. Sin embargo, el costo de procesamiento en el microcontrolador va a ser mayor. Por ello, el *software* permite crear hasta cinco conjuntos difusos para cada entrada y para cada salida. En las figuras 2a y 2b se muestra el diagrama de flujo y el diagrama de actividades para esta sección del programa.

Se espera que el controlador provea una salida para todos los estados de la planta a controlar. A este efecto, todas las variables de entrada y salida del controlador difuso deben poder vincularse a un conjunto difuso. A su vez, cada conjunto difuso debe participar del antecedente o consecuente de una regla. A continuación se describe detalladamente cómo se define la base de reglas.

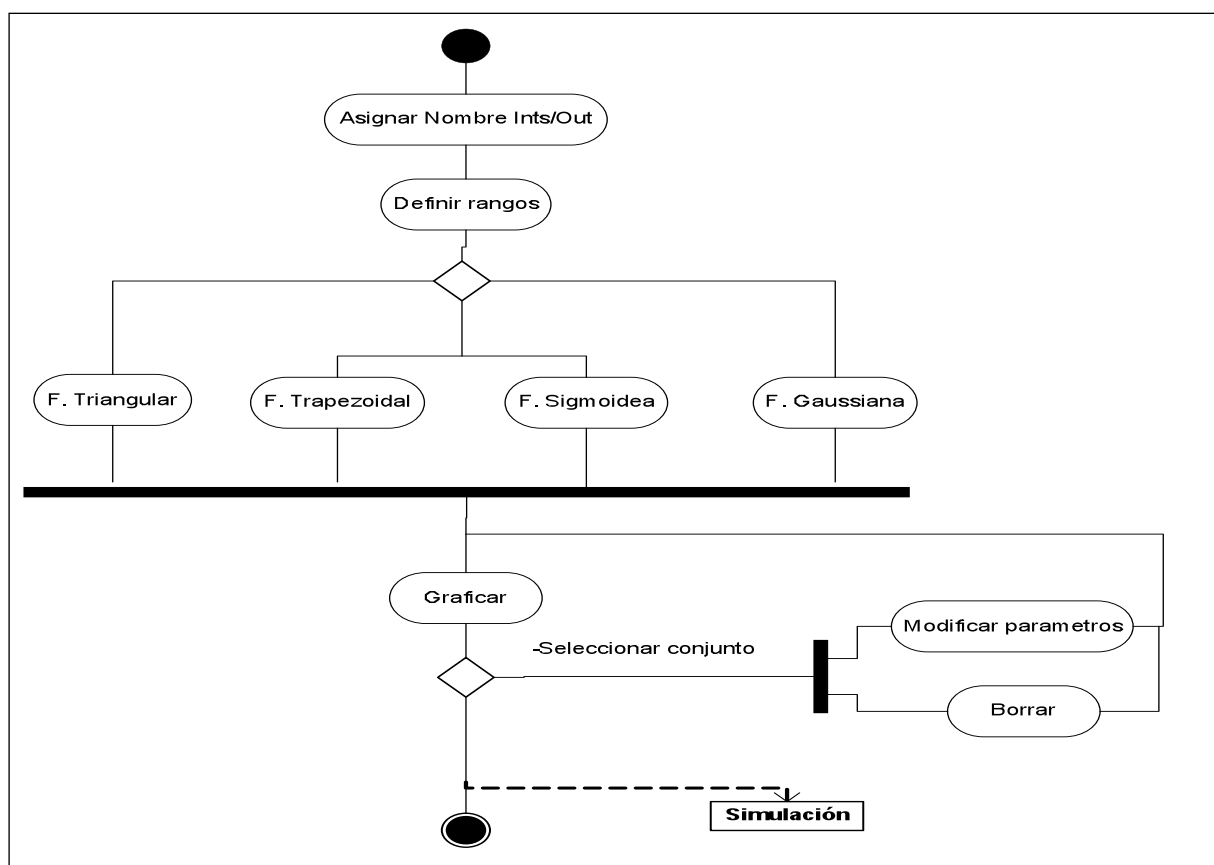


Figura 2 a. Diagrama de actividades uml. Creación y manipulación de conjuntos difusos.

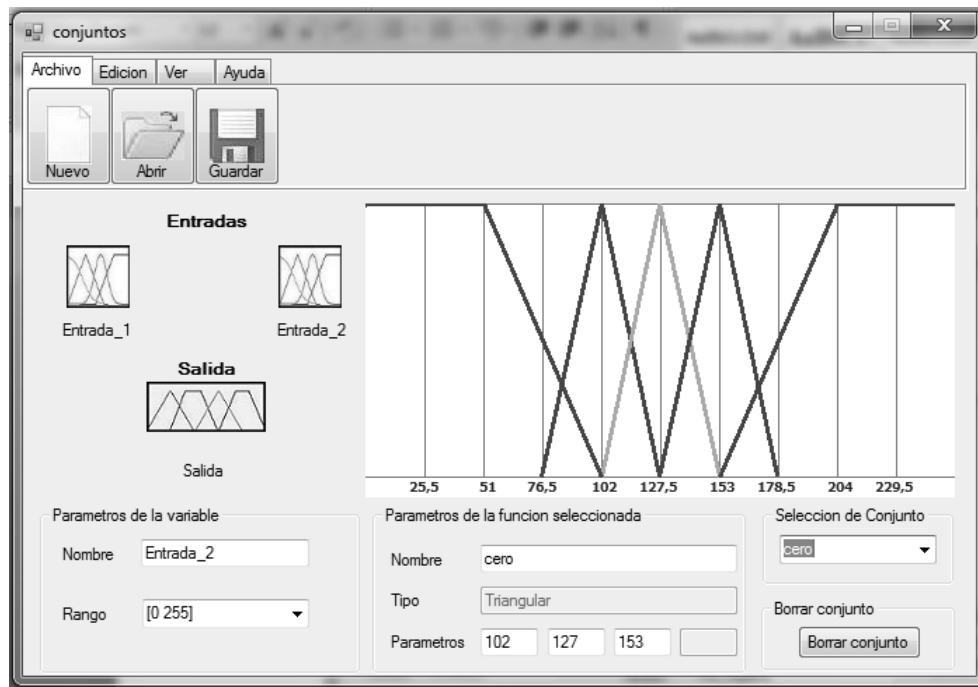


Figura 2b. Formulario de conjuntos Fuzzy Udtectno

Un experto es capaz, mediante un conjunto de reglas que constituyen su experiencia sobre el tema, de elaborar unas conclusiones o consecuentes a partir de unos hechos observados o antecedentes. El experto podrá transmitir sus conocimientos, al menos par-

cialmente, mediante un conjunto de N reglas del tipo si-entonces, con antecedentes relacionados por el conectivo “Y” en la mayoría de los casos, como lo indica la figura 3 [2].

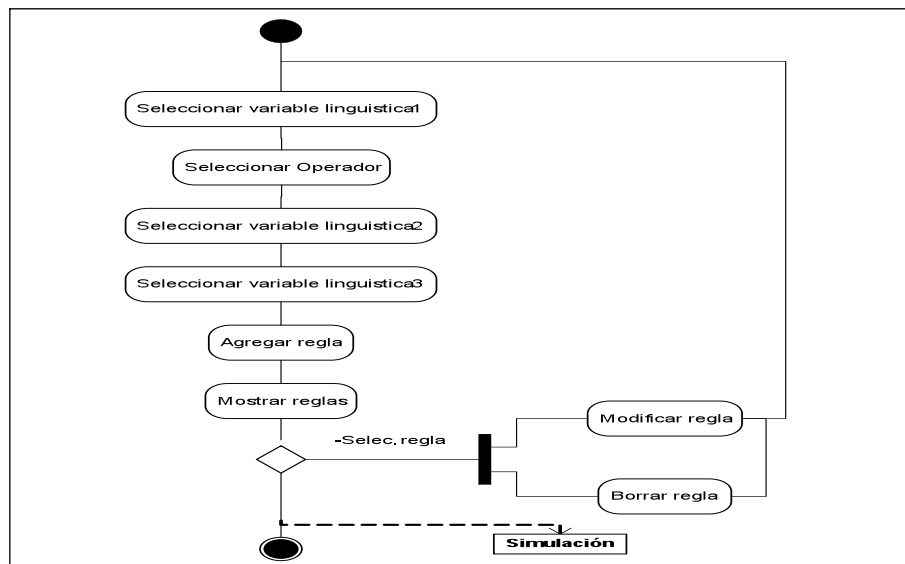
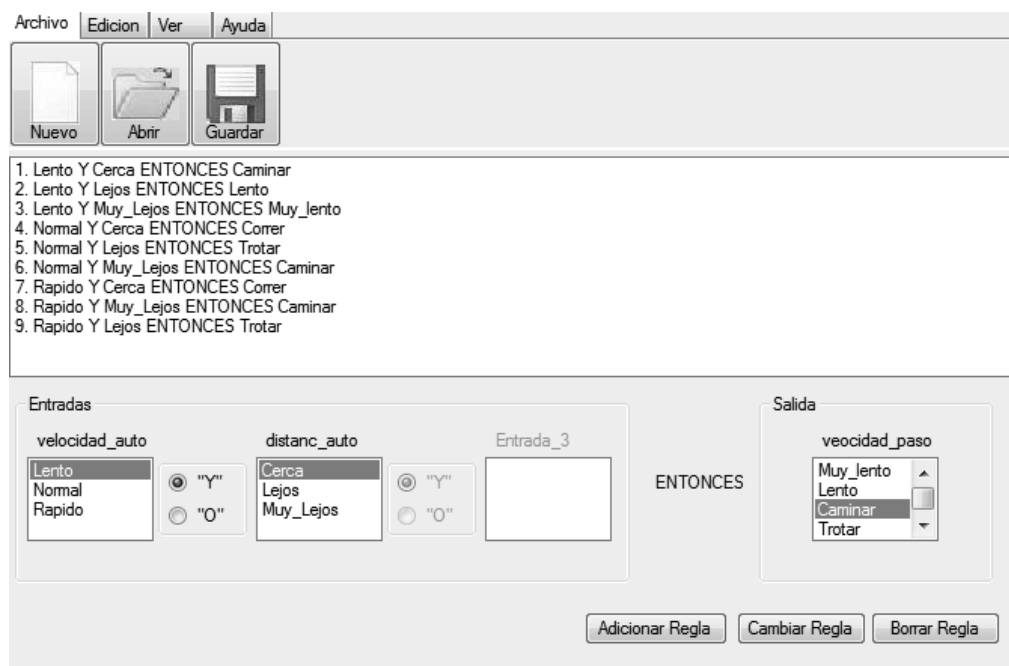


Figura 3a. Diagrama de actividades UML.



3b. Ventana de base de reglas Fuzzy Udteco

En concordancia con las reglas establecidas se procede a diseñar una matriz de inferencia que resume las acciones de control que se tomarán de acuerdo con las diversas condiciones posibles. La matriz se generará para una o dos entradas. Sin embargo, para bases con un mayor número de reglas es necesario hacer una revisión más sistemática. Existen métodos para ello basados en las definiciones de posibilidad y necesidad, a partir de las que puede obtenerse un índice de inconsistencia para cada regla con respecto al conjunto de la base [3].

Una vez están definidas las funciones de pertenencia y la base de reglas, se puede observar el proceso en la ventana de simulación, donde el usuario ingresa valores para cada una de las entradas definidas, se determina en qué grado las variables de entrada pertenecen a los conjuntos difusos –a través de su función de pertenencia–, y se valida cuáles reglas se activan de acuerdo con los

valores de pertenencia. Con las reglas activas se realiza la implicación entre antecedente y consecuente. Para que se cumpla la equivalencia entre lógica proposicional o clásica y lógica difusa [4], se debe traducir por una t-norma, cuyos representantes más utilizados en el ámbito de la ingeniería son el mínimo y el producto.

Una vez evaluadas todas las reglas y obtenidos los conjuntos difusos de salida modificados, hay que realizar la agregación de todas las reglas para obtener un resultado único de la actuación de todas ellas. Esta agregación es una unión lógica, y una vez más, para conservar la equivalencia entre lógicas clásica y difusa, se traduce por una t-conorma: máximo o suma algebraica, obteniéndose así el conjunto difuso de salida.

A partir del conjunto difuso de salida que resulta de la agregación de todas las reglas, se

necesita obtener un resultado escalar, puesto que el actuador necesita tener una magnitud no difusa; las estrategias empleadas más comúnmente son los criterios del promedio de máximos, primer máximo, último máximo y centro de área.

Para generar la superficie, se realiza la defuzzificación para todas las combinaciones de una o dos entradas del sistema. Si se tie-

nen tres entradas, será necesario referenciar una, para así construir la superficie.

Para la generación del código se debe escoger el lenguaje: C18, ANSI C, por el método vector, .fis (MATLAB), y generación de código para memorias I²C (cuando se genera el código para la superficie) en CCS o C18. En la figuras 4 y 5 se muestran los diagramas de flujo para la generación de código en los diferentes lenguajes.

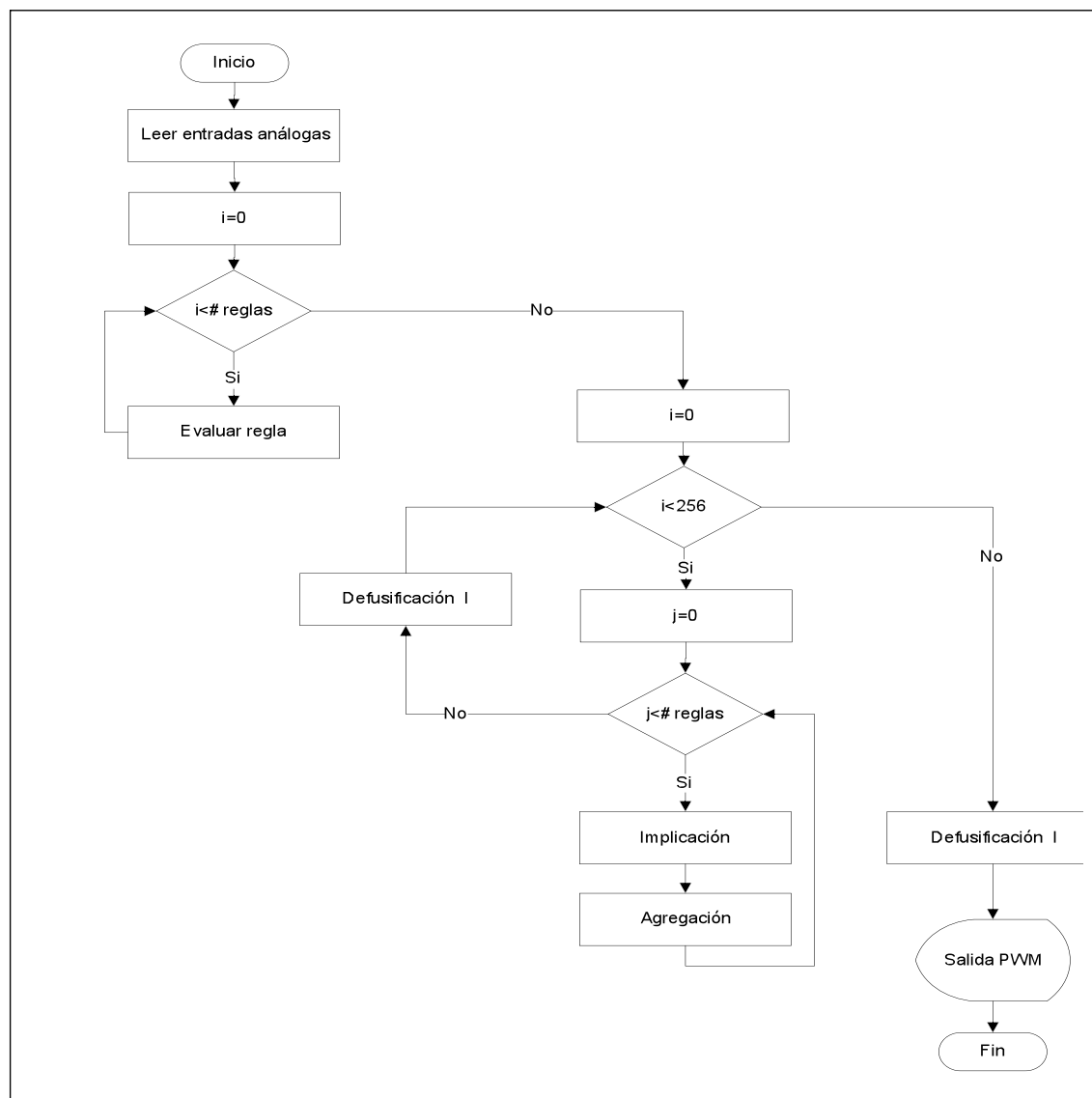


Figura 4. Diagrama de flujo general microcontrolador. Método vector

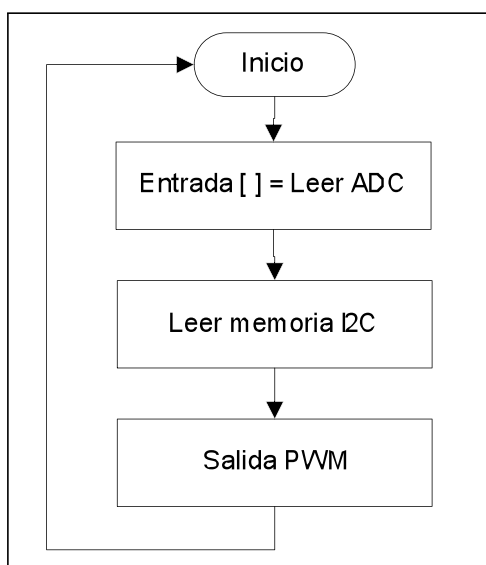


Figura 5. diagrama de flujo general microcontrolador. Memorias I2C

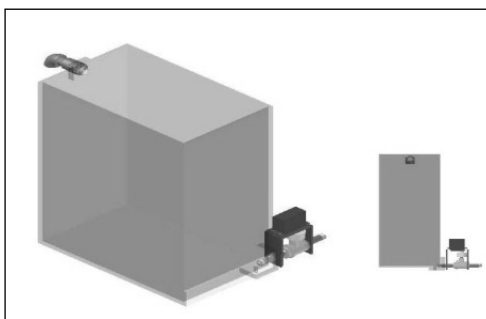


Figura 6. Configuración física de un tanque con desagüe lateral y una servo-válvula al 100% de apertura. Fuente: F. Guzmán e I. Piñeros [5: 68]

2. RESULTADOS

2.1. CONTROLADOR DIFUSO PARA UN SISTEMA MULTITANQUE A ESCALA

Como ejemplo orientador se implementó un controlador difuso para la planta sistema multitank a escala. Se tomó la configuración mostrada en la figura 6.

Es decir, el modelo tanque 1 con capacitancia rectangular modelado en la figura 7.

De la siguiente forma se adecuan las señales de entrada y salida del controlador difuso:

A continuación se diseñó un controlador difuso, para lo cual se tomaron como universos de discurso el error y la derivada del error para las entradas, y el actuador para la salida. Se tiene una implicación de mínimo, agregación de máximo y defuzzificación centro de gravedad. Las funciones de membresía creadas para cada entrada y cada salida, así como la base de reglas, se ilustran en las figuras 9 a 12. En la figura 13 se muestra una simulación en MatLab®.

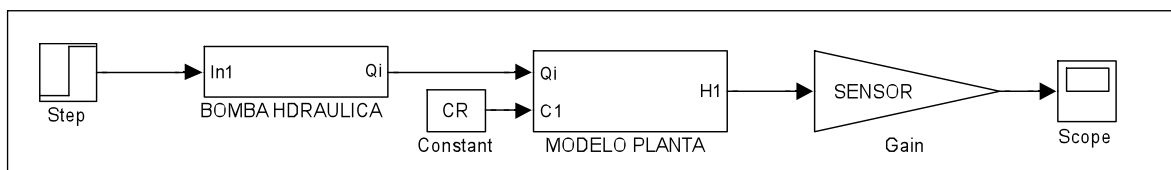


Figura 7. Sistema de bloques multitanque

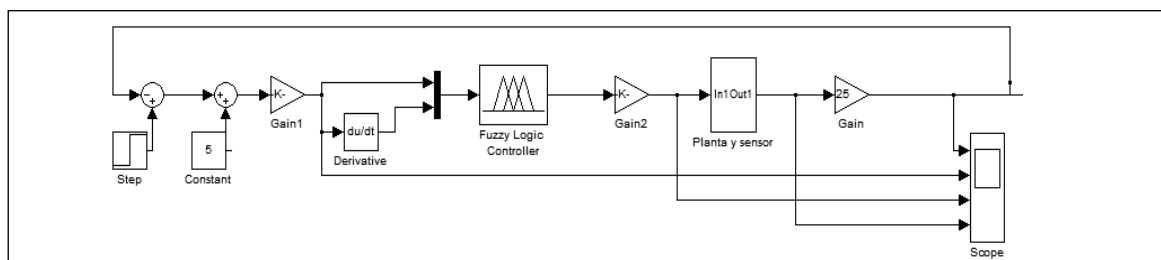


Figura 8. Controlador difuso y planta

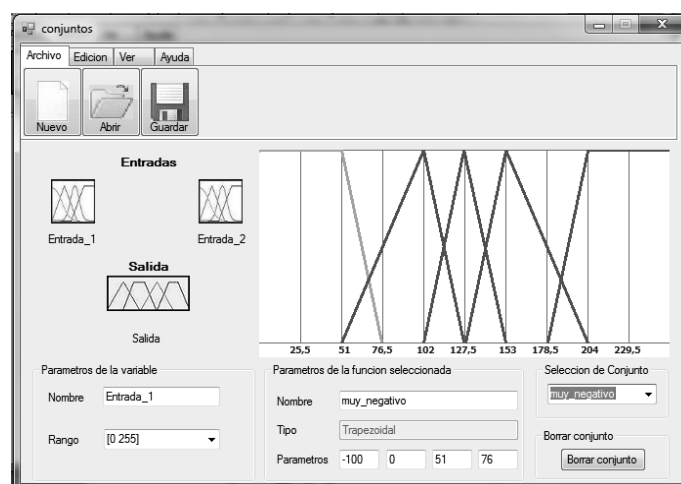


Figura 9. Universo de entrada: error fuzzyudtecno

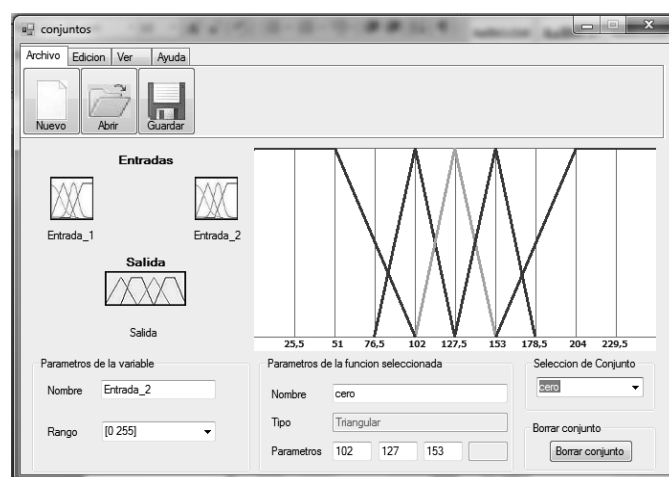


Figura 10. Universo de entrada: derivada del error fuzzyudtecno

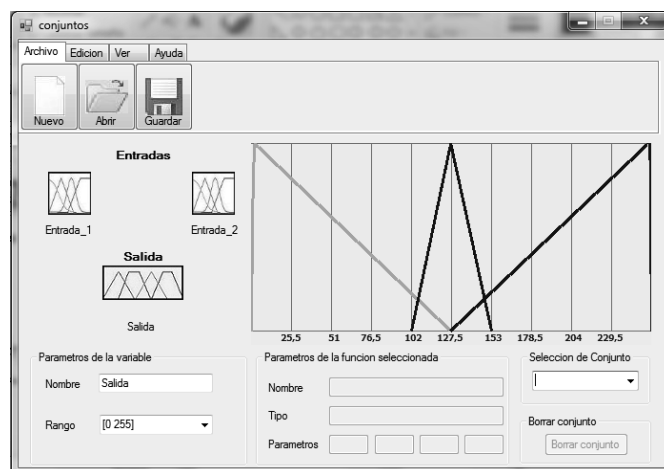


Figura 11. Universo de salida: actuador bomba fuzzyudtecno

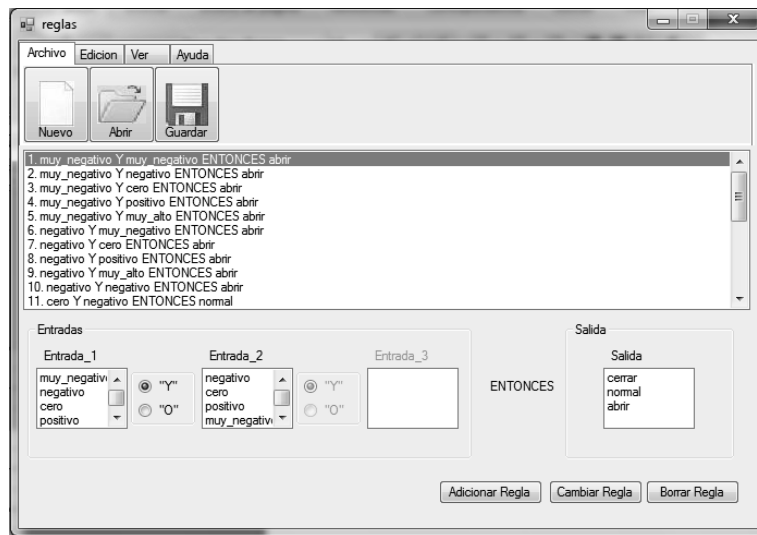


Figura 12. Base de reglas fuzzyudtecno

Si se exporta este controlador al .fis de MAT-LAB se obtiene la siguiente respuesta:

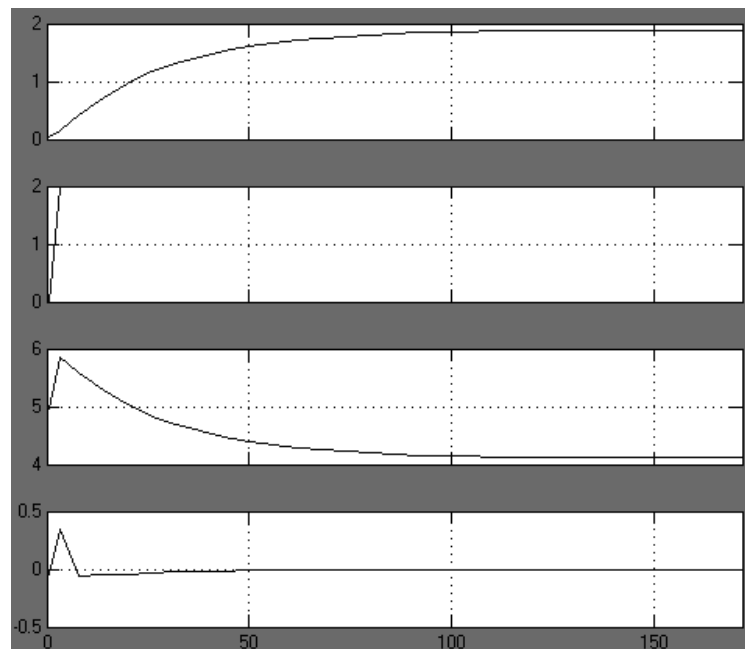


Figura 13. Simulación en MatLab: voltaje vs. tiempo

En la primera figura se observa la señal del sensor; en la segunda, la referencia; el error se observa en la tercera; y por último, en la cuarta, la derivada del error.

2.2. HELICÓPTERO

Para diseñar un controlador difuso no es necesario conocer el modelo matemático de la

planta, basta con tener un conocimiento a fondo del comportamiento del sistema. Este es el caso del helicóptero, en donde se realizó un control para el ángulo de elevación de

acuerdo con su funcionamiento. En las figuras 14 a 16 se muestra la configuración del controlador.

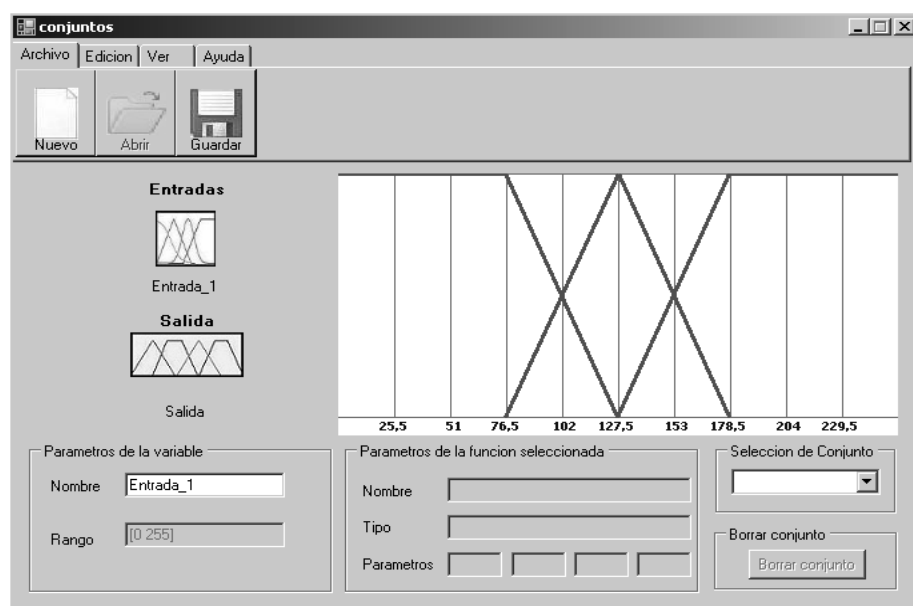


Figura 14. Funciones de membresía para el universo de entrada

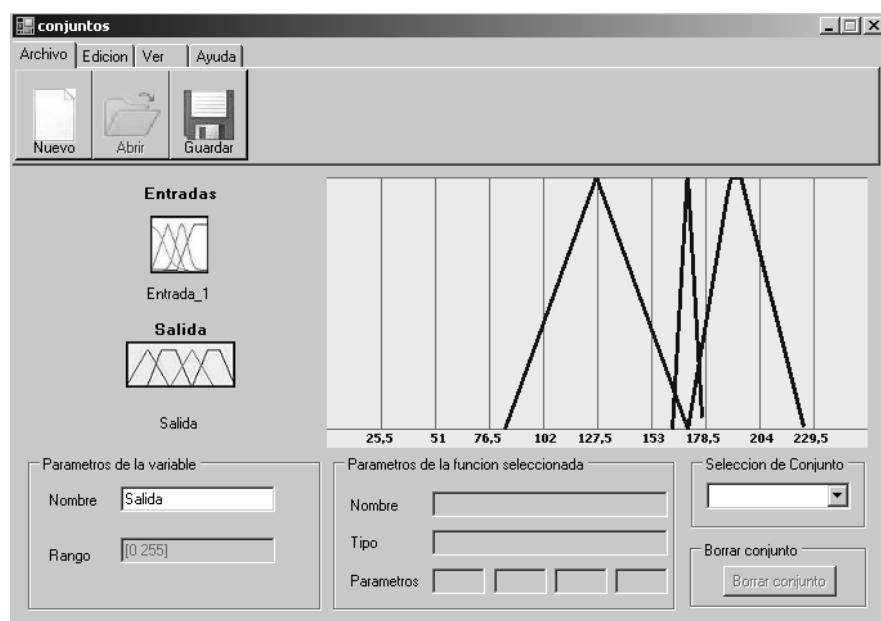


Figura 15. Funciones de membresía para el universo de salida

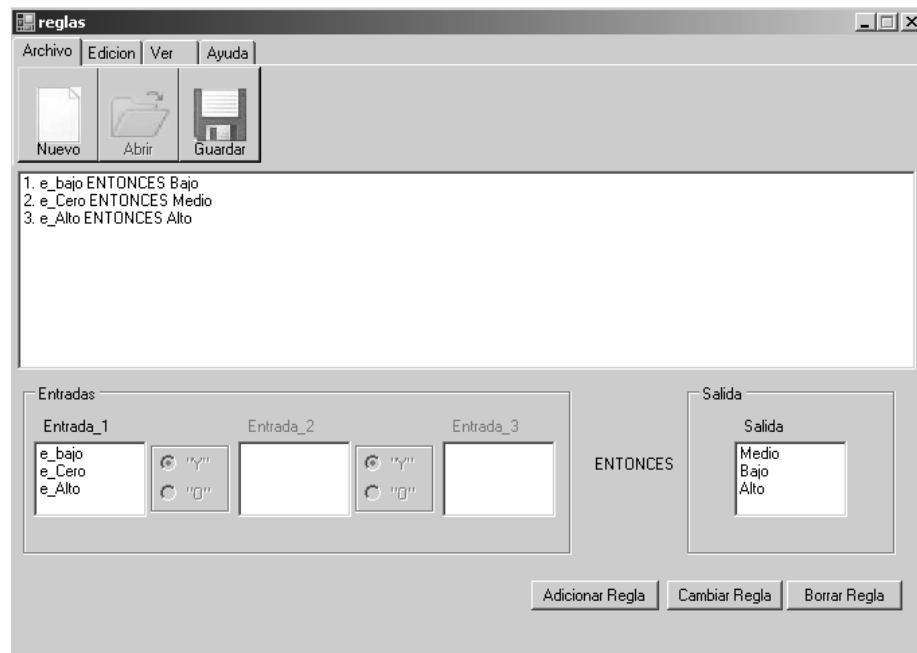


Figura 16. Definición de base de reglas

La gráfica obtenida directamente del sensor y la referencia se muestran en la figura 17.

Es posible, además, diseñar otro controlador del mismo tipo, pero de dos entradas, el cual posee un mejor comportamiento pues trabaja con la derivada del error que ayuda a predecir el comportamiento del controlador.

3. FUTUROS DESARROLLOS

Después de la investigación realizada, quedan expectativas por abordar. Entre los

futuros trabajos a realizar se podrían nombrar los siguientes:

Aplicativo extensionista que permite seleccionar otro tipo de controladores.

Implementación de otro tipo de controladores basados en lógica difusa, tales como Takagi Sugeno, o lógica difusa tipo II. Comunicación RS232, e incluso Ethernet. Implementación de algoritmos de optimización para reducir el tamaño de datos en el caso de las superficies.

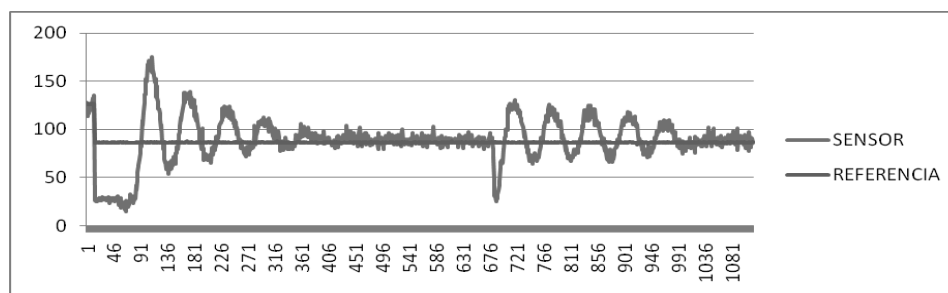


Figura 17. Señal tomada de la planta

4. CONCLUSIONES

El trabajo con sistemas difusos aplicados al control implica el desarrollo de librerías para su implementación. Luego de hacerlo, se resuelve gran parte de los obstáculos en la aplicación en dispositivos tales como microcontroladores o microprocesadores.

La implementación de sistemas difusos o de controladores difusos en microcontroladores tiene un elevado coste computacional, debido a las múltiples operaciones difusas a realizar. Este trabajo concluye que la aplicación que utiliza la superficie difusa en una memoria de estado sólido, reduce el tiempo de cálculo y simplifica el sistema de control a una búsqueda simple en la memoria.

La herramienta desarrollada simplifica y reduce el proceso de diseño e implementación de sistemas difusos, lo que permite al estudiante probar el diseño en un corto período de tiempo, ajustar el sistema y verificar el funcionamiento.

Este recurso permitirá a los estudiantes de los niveles técnico y tecnológico, la apropiación y el uso de la tecnología difusa en el diseño y aplicación de los controladores, objetivo que es vigente, si se tiene en cuenta la creciente utilización e implementación de los conceptos difusos para la solución de problemas de ingeniería.

Referencias

- [1] G. Meneses, "Control difuso de temperatura para carga resistiva con microcontrolador de 8 bits", tesis de Especialización en Ciencias Electrónicas e Informática: Área Automatización, Universidad de Antioquia, pp. 3-4, 2005.
- [2] L.A. Zadeh, "Fuzzy Logic, Neural Networks and Soft Computing", Communications of the ACM, vol. 3, no. 3, pp. 77-84, 1994.
- [3] Pederycz, W., Fuzzy Control and Fuzzy Systems, John Wiley & Sons, pp. 65-84, 1998.
- [4] L.A. Zadeh, "Fuzzy Logic, Neural Networks and Soft Computing", Communications of the ACM, vol. 3, no. 3, pp. 88-94, 1994.
- [5] F. Guzmán e I. Piñeros, "Análisis matemático e implementación de un sistema multitanque a escala para docencia e investigación en un laboratorio de control en la facultad tecnológica", Ingeniería en Control Electrónico, pp. 40-107, Bogotá: Universidad Distrital Francisco José de Caldas, Facultad Tecnológica, 2009.
- [6] K. Passino y S. Yurkovich, Fuzzy Control. Addison-Wesley, pp. 134-45, 1998.
- [7] J.M. Mendel, "Fuzzy Logic Systems for Engineering: A Tutorial", Proceedings of the IEEE, vol. 83, no. 3, pp. 345-77, 1995.